

WHAT IS CLAIMED IS:

1 1. An apparatus for executing at least one single
2 program multiple data (SPMD) program in a microprocessor,
3 said apparatus comprising:

4 a micro single instruction multiple data (SIMD) unit
5 located within said microprocessor; and

6 a job buffer having an output coupled to an input of
7 said micro SIMD unit;

8 wherein said job buffer dynamically allocates tasks to
9 said micro SIMD unit.

1 2. The apparatus as set forth in Claim 1 wherein
2 said micro SIMD unit is capable of sending job status
3 information to said job buffer.

1 3. The apparatus as set forth in Claim 1 wherein
2 said at least one SPMD program comprises a plurality of
3 input data streams having moderate diversification of
4 control flows.

1 4. The apparatus as set forth in Claim 3 wherein
2 said apparatus executes said at least one SPMD program once
3 for each input data stream of said plurality of input data
4 streams.

1 5. The apparatus as set forth in Claim 4 wherein
2 said apparatus generates an instruction stream for each
3 input data stream of said plurality of input data streams.

1 6. The apparatus as set forth in Claim 3 wherein
2 said apparatus executes a plurality of SPMD programs and
3 wherein each SPMD program of said plurality of SPMD
4 programs is executed on a number of input data streams.

1 7. The apparatus as set forth in Claim 6 wherein
2 said number of input data streams is greater than a program
3 granularity threshold.

1 8. The apparatus as set forth in Claim 1 wherein
2 said job buffer dynamically allocates tasks to said micro
3 SIMD unit by dynamically bundling jobs to be executed based
4 on a control flow equivalence of said jobs.

1 9. The apparatus as set forth in Claim 8 wherein
2 said apparatus performs job clustering to form a job bundle
3 in which each job in said job bundle has an equivalent
4 control flow.

1 10. The apparatus as set forth in Claim 9 wherein
2 said apparatus performs said job clustering based on a job
3 processing status of said jobs in said job bundle.

1 11. The apparatus as set forth in Claim 8 wherein
2 said apparatus forces a task to terminate at a point where
3 a job control path might fork by placing a code-stop in
4 said task.

1 12. The apparatus as set forth in Claim 11 wherein
2 said apparatus minimizes a required number of code-stops to
3 be placed in said task by excluding from code-stop
4 placement each control flow statements that is equivalent
5 to a select instruction.

1 13. The apparatus as set forth in Claim 9 wherein
2 said apparatus maximizes a size of a job cluster by
3 selecting tasks for execution in which a job processing
4 status of each of said tasks is complete.

1 14. The apparatus as set forth in Claim 8 wherein
2 said apparatus executes a data loading phase for a task
3 before said apparatus executes a task execution phase for
4 said task.

1 15. A method for executing at least one single
2 program multiple data (SPMD) program in a microprocessor,
3 said method comprising the steps of:

4 providing a micro single instruction multiple data
5 (SIMD) unit located within said microprocessor;

6 providing a job buffer having an output coupled to an
7 input of said micro SIMD unit; and

8 dynamically allocating tasks to said micro SIMD unit
9 in said job buffer.

1 16. The method as set forth in Claim 15 further
2 comprising the step of:

3 sending job status information from said SIMD unit to
4 said job buffer.

1 17. The method as set forth in Claim 15 wherein said
2 at least one SPMD program comprises a plurality of input
3 data streams having moderate diversification of control
4 flows.

1 18. The method as set forth in Claim 17 further
2 comprising the step of:

3 executing said at least one SPMD program once for each
4 input data stream of said plurality of input data streams.

1 19. The method as set forth in Claim 18 further
2 comprising the step of:

3 generating an instruction stream for each input data
4 stream of said plurality of input data streams.

1 20. The method as set forth in Claim 17 further
2 comprising the steps of:

3 executing a plurality of SPMD programs; and
4 executing each SPMD program of said plurality of SPMD
5 programs on a number of input data streams.

1 21. The method as set forth in Claim 20 wherein said
2 number of input data streams is greater than a program
3 granularity threshold.

1 22. The method as set forth in Claim 15 wherein said
2 job buffer dynamically allocates tasks to said micro SIMD
3 unit by dynamically bundling jobs to be executed based on a
4 control flow equivalence of said jobs.

1 23. The method as set forth in Claim 22 further
2 comprising the step of:

3 performing job clustering to form a job bundle in
4 which each job in said job bundle has an equivalent control
5 flow.

1 24. The method as set forth in Claim 23 further
2 comprising the step of:

3 performing said job clustering based on a job
4 processing status of said jobs in said job bundle.

1 25. The method as set forth in Claim 22 further
2 comprising the step of:

3 forcing a task to terminate at a point where a job
4 control path might fork by placing a code-stop in said
5 task.

1 26. The method as set forth in Claim 25 further
2 comprising the step of:

3 minimizing a required number of code-stops to be
4 placed in said task by excluding from code-stop placement
5 each control flow statements that is equivalent to a select
6 instruction.

1 27. The method as set forth in Claim 23 further
2 comprising the step of:

3 maximizing a size of a job cluster by selecting tasks
4 for execution in which a job processing status of each of
5 said tasks is complete.

1 28. The method as set forth in Claim 22 further
2 comprising the step of:

3 executing a data loading phase for a task before
4 executing a task execution phase for said task.